新連載

実設計に即した車載LAN

FlexRay 仕様解説

第1回 トポロジとメディア・アクセス方式

システムの記事

佐藤道夫

本連載では，2004年7月に発表されたFlexRayプロトコルのバージョン2.0をもとに，仕様の解釈や実装方法を解説する．FlexRayに対応したネットワーク・システムを開発する際の重要な項目として，トポロジ，ノード数，リアルタイム性，ネットワーク速度，フォールト・トレラント機能，最大データ・サイズ，フレーム形式が挙げられる．今回はまず，トポロジとリアルタイム性について考える．（編集部）

FlexRayコンソーシアムが2000年に発足してから約4年がたちました．FlexRayは，「車載LANの高速化」という市場要求に対応するために策定されている規格です．データ転送速度は最大10Mbpsとされており，現在パワートレイン系に採用されている500kbpsのCAN-Cを置き換えることが予想されます．また，ドイツのBMW社やDimler Chrysler社，米国General Motors社など，世界の大手自動車メーカがコンソーシアムに参加したこともあり，これからの車載制御ネットワーク・プロトコルとして注目されています．とくに日本では，トヨタ自動車や日産自動車，本田技研工業，マツダなどの主だった自動車メーカが参加し，さらにFlexRayの標準化が加速しました（**表1**）．

筆者は，機会があるごとにFlexRayプロトコルの概要を本誌や各種の技術セミナなどでお話ししてきました．2004年7月に最新のバージョン2.0が公開されたことで，ようやく実装まで含めた詳細を話せるようになりました．本連載では，このバージョン2.0の内容を中心に，実設計を視野に入れた仕様の解釈と実装方法について説明していきたいと思います．

本連載が，実際に仕様書を読んで設計に取りかかる際の

表1 FlexRayコンソーシアムの組織

2005年2月中旬現在のメンバ．2003年末から2004年の初めにかけて，日本を代表する自動車メーカであるトヨタ自動車や日産自動車，本田技研工業がプレミアム・アソシエイツ・メンバとして入会した．今後は各自動車メーカがどのようなアプリケーションでFlexRayを搭載するかが注目される．

コア・メンバ（7社）	BMW社，Bosch社，DaimlerChrysler社，Freescale Semiconductor社，General Motors社，Philips社，Volkswagen社
プレミアム・アソシエイツ・メンバ（13社）	Continental Teves社，Delphi社，デンソー，Fiat社，Ford Motor社，本田技研工業，Hyundai社，マツダ，日産自動車，PSA Peuqeot Citroen社，Renault社，トヨタ自動車，Tyco Electronics社
アソシエイツ・メンバ（53社）	アドバンスド・データ・コントロールズ，アイシン精機，Alpine Electronics社，AMI Semiconductor社，Atmel社，austriamicrosystems社，Avidyne社，BERATA社，BerTrandt社，カルソニックカンセイ，EADS Deutshland社，Elmos Semiconductor社，ESG社，Esterel Technologies社，Eurospace社，富士通，富士通テン，Haldex社，Hella社，日立電線，日立製作所，Hyundai Autonet社，IAV社，Infineon Technologies社，IPETRONIK社，iRoC Technologies社，三菱電機，Molex Holding社，NECエレクトロニス，NSK ステアリングシステムズ，沖電気工業，Pacifica Group Technologies社，Porsche社，ルネサス テクノロジ，RWTÜV社，SCANIA社，Siemens VDO Automotive社，SP社，STMicroelectronics社，富士重工業，住友電工，スズキ，TDK，Texas Instruments社，ThyssenKrupp Automotive Mechatronics社，TNI Software社，豊田通商，TRW Conekt社，Visteon社，Würth Elektronik社，ヤマハ発動機，矢崎，横河電機
デベロッパ・メンバ（31社）	3SOFT社，Berner & Mattner Systemtechnik社，C&S社，Cadence Design Systems社，CANway Technology社，CapeWare社，Cardec社，CRST社，DECOMSYS社，dSPACE社，ETAS社，FIZ社，GIGATRONIK社，GÖPEL electronic社，Intrepid Control Systems社，IXXAT社，Lautebach Datentechnik社，MicroSys社，Mission Level Design社，National Instruments社，NSI社，SEDES Special Electronic Design社，Softing社，SystemA Engineering社，TECWINGS社，トヨタマックス，TZM社，Vector社，Volcano Automotive社，Warwick Control Technologies社，Weise社

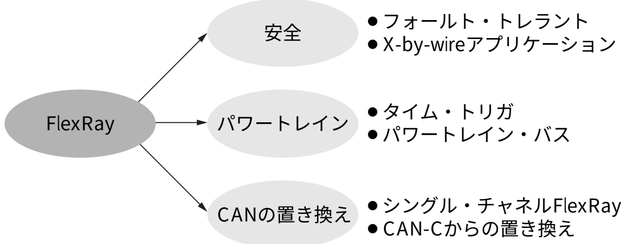


図1　FlexRayアプリケーションの方向性
FlexRayには三つの方向性がある．すなわち，安全系アプリケーション，パワートレイン制御系アプリケーション，CANの置き換えである．

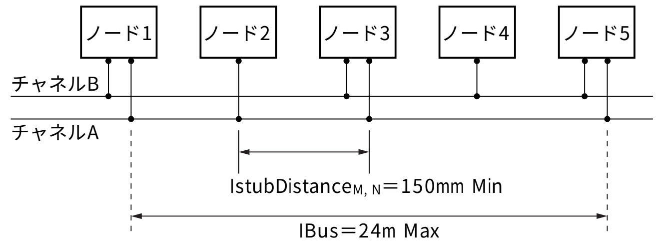


図2　バス型トポロジ
バス型では，1チャネル当たり，ノード間のポイント・ツー・ポイントのバスのワイヤ長は最大24m（IBus＝24m Max），ノード数は最大22ノード（bStub Nodes＝22 Max），二つのノードの距離は最小150mm（$IstubDistance_{M, N}$＝150mm Min）に規定されている．

一助になれば幸いです．

1 仕様の構成と方向性

FlexRayプロトコルのバージョン2.0はFlexRayコンソーシアムのWebサイト（http://www.flexray-group.org/）からダウンロードできます．プロトコル仕様書は全10章に加え，付録A，Bからなっており，200ページ強のボリュームがあります．また，仕様の記述はSDL（Specification and Description Language）をベースとしています．なお，FlexRayでは本連載で取り上げるプロトコル仕様書のほかに，「電気的物理層仕様書」と「バス・ガーディアン仕様書」も発行されています．

本連載では，FlexRayの仕様書について極力日本語に訳して説明しますが，中には英語の表現をそのまま流用することがあります．とくにパラメータはそのまま表記していきます．これは，設計時には基本的に英語版の仕様書が基準になるので，ここで取り上げる内容や表現が読者のみなさんに誤解を与えることがないようにするためです．

なお，本連載ではプロトコル仕様書の内容をすべて取り上げるわけではありません．設計するうえで基本となる項目や，仕様書を読んだだけではわかりにくい項目などを重点的に解説していきます．

また，通信コントローラLSIへの実装例も紹介する予定です．プロトコル仕様書と対比しながら，FlexRayプロトコルがどのように動作するのかを理解していただきたいと思います．

● FlexRayのターゲット・アプリケーション分野は三つ

現在，自動車は世界的に普及しています．このため，環境保護の観点からより精密な制御が要求されるようになりました．これを実現するために，自動車内の制御は機械的制御から電子的制御へと着実に移行しています．また，安全面から新しいアプリケーションが登場し始めています．例えば，エア・バッグに代表される衝突時における乗員の傷害を軽減することを目的とした「パッシブ・セーフティ」に加えて，自動車の衝突そのものを回避することを目的とした「アクティブ・セーフティ」が搭載されるようになりました．

FlexRayは，**図1**に示すように現在のところ三つの方向性が考えられています．最初に市場に投入されるのは，おそらくCAN-Cの置き換えのアプリケーションだと思われます．

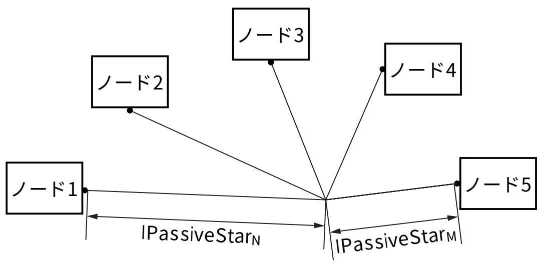
2 ネットワーク設計の要件を決める

ネットワークのシステム設計に必要な主な項目を以下に挙げます．このほかにもいろいろありますが，これでクラスタ[注1]の大まかな仕様を決めることができます．

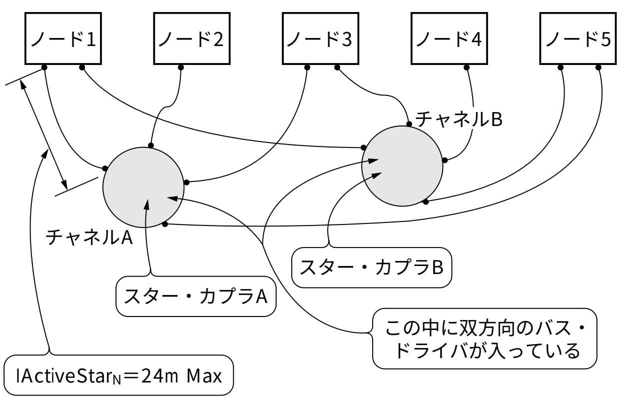
- トポロジ
- ノード数
- リアルタイム性
- ネットワーク速度
- フォールト・トレラント機能
- 最大データ・サイズ
- フレーム形式

こうした項目について，FlexRayネットワークではどのように仕様が決められているのでしょうか．それを本連載

注1：クラスタは，複数のノード（ECU：electronic control unit）の集まりを指す．



（**a**）パッシブ・スター型



（**b**）アクティブ・スター型

図3　スター型トポロジ
スター型のトポロジには，パッシブ・タイプとアクティブ・タイプの2種類がある．

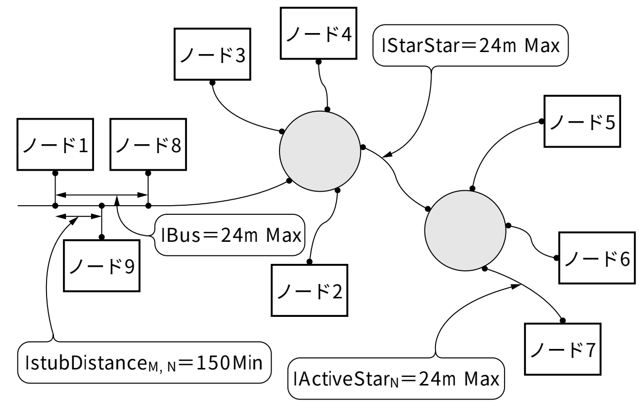
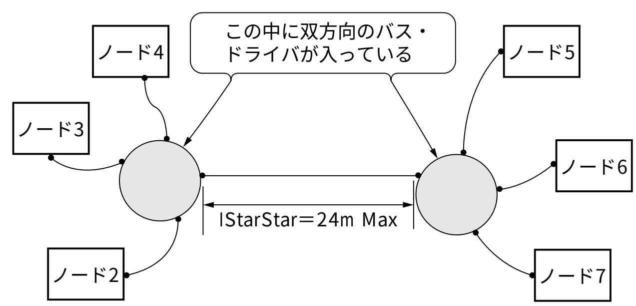


図4　バスとスターの混載型のトポロジ
このトポロジの電気的物理層は，前述したバス型とスター型の仕様に準じる．



（**c**）カスケード・アクティブ・スター型

では詳しく説明していきます．

● トポロジは「バス型」，「スター型」，「混載型」に対応

自動車の場合，大型車，中型車，小型車，商用車，乗用車などさまざまな形状があります．ここにネットワークのワイヤを設置しようとすると，柔軟性が要求されます．この要求にこたえるため，FlexRayのトポロジは，バス型（**図2**），スター型（**図3**），バスとスターの混載型（**図4**）に対応しています．

電気的物理層の仕様では，バス型では1チャネル当たりのノード間のポイント・ツー・ポイントのバスのワイヤ長は最大24m（IBus＝24m Max），ノード数は最大22ノード（bStubNodes＝22 Max）となっています．また，隣り合う二つのノードの距離は最小150mm（$\mathrm{IstubDistance_{M,\,N}}$＝150mm Min）に規定されています．

スター型には，パッシブ・タイプとアクティブ・タイプの2種類があります．パッシブ・スター型は基本的にバス型と同じ考えかたですが，$\mathrm{IstubDistance_{M,\,N}}$がゼロになったものをFlexRayの仕様ではパッシブ・スター型と位置づけています．パッシブ・スター型でポイント・ツー・ポイント接続を考えた場合，ワイヤ長は最大24m（$\mathrm{IPassiveStar_N}$＋$\mathrm{IPassiveStar_M}$=24m Max），ノード数は最大22ノード（nStarNode＝22 Max）です．

一方，アクティブ・スター型は，**図3（b）**に示すように各ノードを「スター・カプラ」と呼ばれる装置で接続しています．ワイヤ長は最大24m（$\mathrm{IActiveStar_N}$＝24m Max）ですが，これはノード間ではなく，ノードとスター・カプラの距離です．アクティブ・スター型にはバス・ドライバ機能が備わっており，これによって最大24mのワイヤ長を実現しています．また，アクティブ・スターに接続されるノード数はバス・ドライバの数に依存するため，最大接続数はとくに規定されていません（詳細は，電気物理層仕様書第9章を参照のこと）．

図3（c）は，アクティブ・スターを複数接続した例です．FlexRayではカスケード接続が仕様化されており，最大二つ（$\mathrm{nStarPath_{M,\,N}}$＝0 Min，2 Max）のアクティブ・スタ

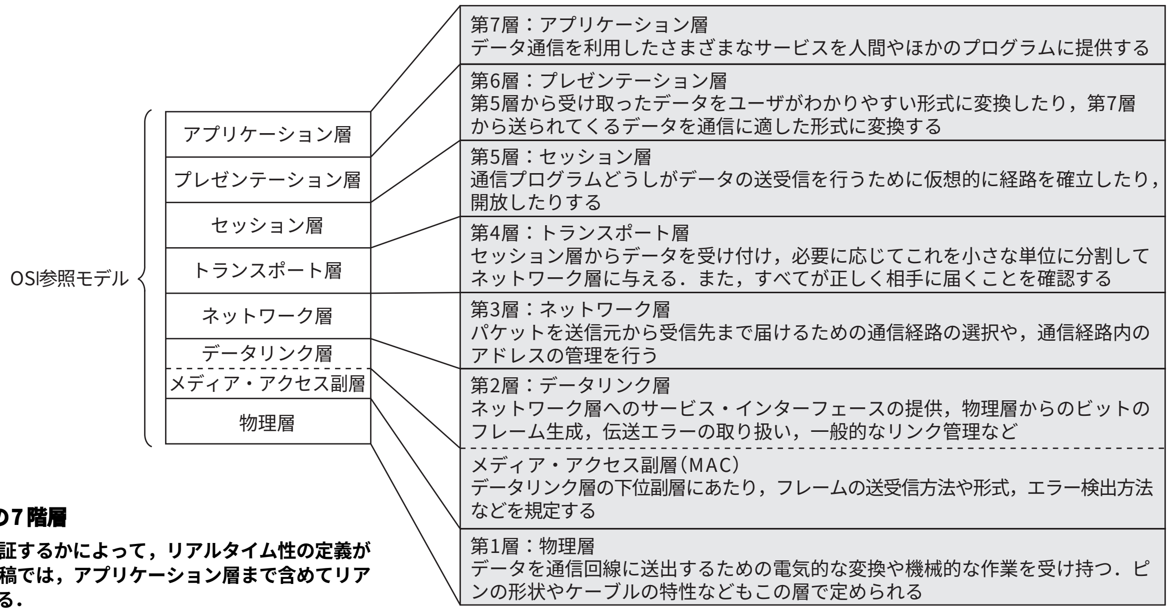


図6
OSI 参照モデルの7 階層
どの階層までを保証するかによって，リアルタイム性の定義が異なってくる．本稿では，アプリケーション層まで含めてリアルタイム性を考える．

ーをカスケード接続できます．アクティブ・スター間のワイヤ長は最大24m（IStarStar＝24m Max）です．

図4のバスとスターの混載型のトポロジにおける電気的物理層は，前述したバス型とスター型の仕様に準じます．

ちなみに，FlexRayプロトコル仕様書では1クラスタの最大ノード数は64ノード（cController Max＝64）と規定されています．プロトコル仕様書と電気的物理層仕様でノード数が異なるのは，一つにはプロトコルが物理層に依存しないように仕様化されているからです．例えば，今後64ノードを駆動できるバス・ドライバが出現したとしても，現在のプロトコルでこれを実現できるものと思います．

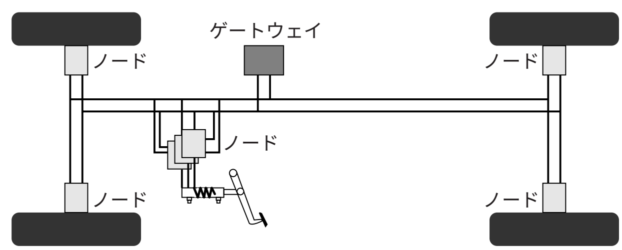


図5　Brake-by-wireのアプリケーション例
X-by-wire技術は，パワートレイン系を従来の機械的制御から電子的制御へ転換させる．X-by-wireのアプリケーションの一つとして，油圧ポンプではなく電気信号を用いてブレーキ制御を行うBrake-by-wireがある．図の中で，薄い灰色の四角はペダルやタイヤを制御するノード（ECU）を，濃い灰色の四角はゲートウェイを表す．

● リアルタイム性を理解するにはまずアクセス方式から

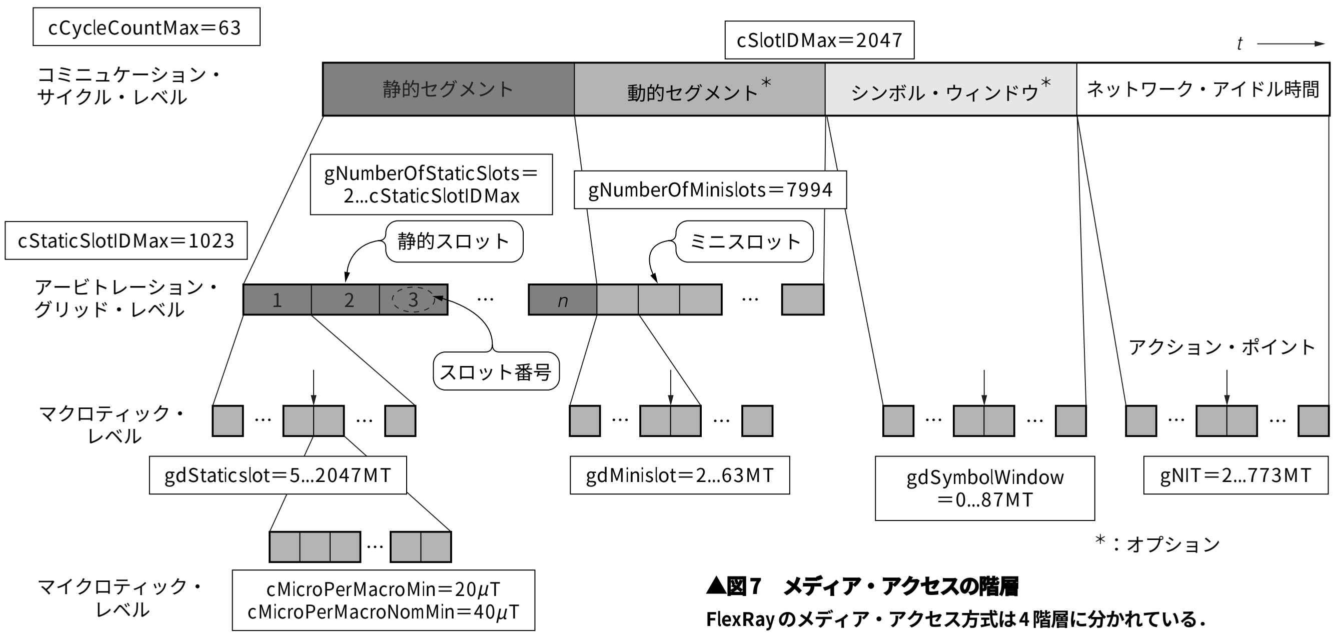
次に，リアルタイム性について考えましょう．ここでは，ブレーキ・システムを例に挙げて説明します（**図5**）．

運転者がブレーキ・ペダルを踏み込んで自動車が停止するまで，あるいは運転者の所望する速度になるまでが，ブレーキ・システムのリアルタイム性と考えることができます．なお，本連載ではリアルタイム性を「アプリケーション層の実行からアプリケーション層の実行までの時間」と定義します．このネットワークのリアルタイム性については，OSI（Open System Interconnection）参照モデルの7階層のどこまでを保証するかで考えかたが変わってきます（**図6**）．制御対象によってアルゴリズムが異なるため，アプリケーションの実行時間は変化に富んでいます．そのため，ネットワーク・プロトコルのリアルタイム性を議論するとき，アプリケーション層を含めないこともあります．しかし，実使用を考えると，アプリケーション層までを含めることによってネットワーク速度やクラスタの規模などの仕様がおのずと決まるものだと筆者は考えます（なお，ネットワークのリアルタイム性を考えるとき，パケットが確実に送受信できるかどうかについて検討する必要があるが，これについては次回以降に解説する）．

さて，このリアルタイム性を考えるうえで重要になるのがメディア・アクセス方式です．これを十分理解しておかなくてはなりません．

1）コミュニケーション・サイクル・レベル

図7に示すように，FlexRayのメディア・アクセス方式は4階層に分かれています．最大のアクセス単位はコミュ



▲図7　メディア・アクセスの階層
FlexRayのメディア・アクセス方式は4階層に分かれている．

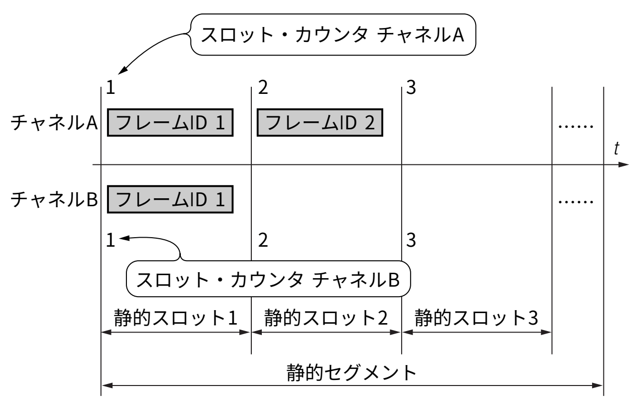


図8　静的セグメントの構造
スロット番号とスロット・カウンタの状態変数が一致すると，フレームを送信する．

ニケーション・サイクルです．あるノードが一度データを送信し，再度データを送信するためのスロットを得るまでの時間はこのコミュニケーション・サイクルで決まります．まず，コミュニケーション・サイクルをどのくらいの時間にするのかを考えなくてはなりません．

例えば，**図5**のブレーキ・システムは八つのノードで構成されているとします．その内訳をペダル・ノードが三つ，車輪のノードが四つ，ゲートウェイ・ノードが一つとしましょう．そして，このブレーキ・システムのコミュニケーション・サイクルを1msと決めます．これは言い換えると，1msごとにペダルのステータス（状況）を四つの車輪ノードに送信し，各車輪のステータスも最悪1ms以内に把握できるようにするということです．

FlexRayのプロトコル仕様では，コミュニケーション・サイクル時間は最大16ms（cdCycleMax＝16,000μs）となっています．また，1コミュニケーション・サイクルにおけるマクロティック（詳細は後述）の設定範囲は12～16,000（gMacroPerCycle（12...16,000MT））と規定されています．各ノードはこのコミュニケーション・サイクルのカウンタを持っています．このカウンタの最大値は63（cCycle CountMax＝63）です．この最大値を超えると0に戻り，再度カウントアップします．

コミュニケーション・サイクル・レベルは，静的セグメント，動的セグメント，シンボル・ウィンドウ，ネットワーク・アイドル時間から構成されています．静的セグメントと動的セグメントは送信フレームのスケジューリングを行います．また，シンボル・ウィンドウは，スタートアップ時，ウェイク・アップ時などでオプションとして使われます．ネットワーク・アイドル時間は重要な役割を持っており，クロック同期のオフセットや伝送速度の計算，エラー訂正などに使用されます．なお，動的セグメントとシンボル・ウィンドウはオプションになっています（pDynamicSegmentEnable＝Boolean[注2]，gdSymbolWindow＝0...87 MT）．

注2：pDynamicSegmentEnable＝Booleanは，Boolean＝1（真）であれば動的セグメントを使い，Boolean＝0（偽）であれば動的セグメントを使わないということを意味する．gdSymbolWindow（シンボル・ウィンドウ）の場合，0のときはシンボル・ウィンドウはなし．

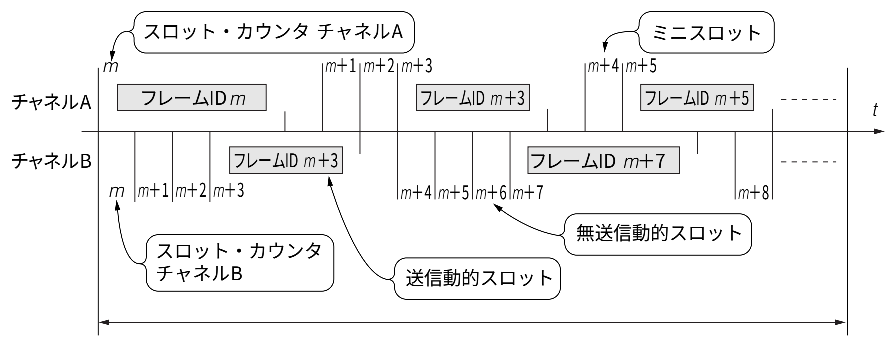


図10　動的セグメントの構造

フレームIDとスロット・カウンタの状態変数が一致すると，フレームを送信する．なお，動的セグメントは最大7,994のミニスロット（gNumberOfMinisolots=7994）を含んでいる．

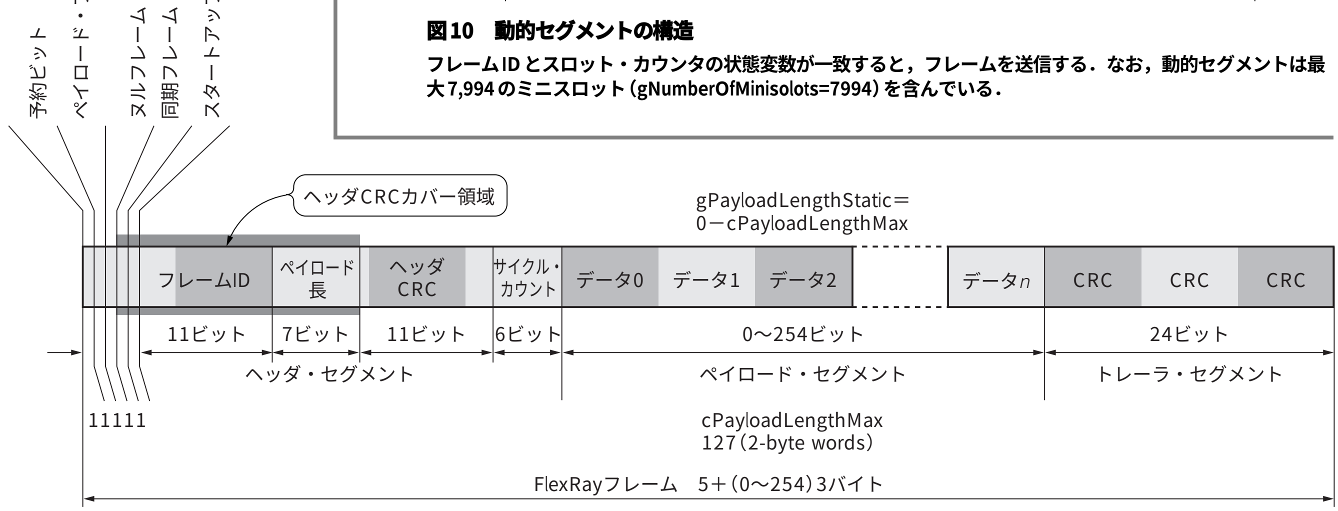


図9　FlexRayのフレーム・フォーマット

2）アービトレーション・グリッド・レベル

次は，アービトレーション・グリッド・レベルです．このレベルは，衝突のないメディア・アクセスを実現するためのものです．構成要素としては，静的スロットとミニスロットがあります．

図8に，静的セグメントの構造を示します．図のように，送信フレームはあらかじめスケジューリングされています．チャネルAとチャネルBの各ノードは，スロット・カウンタの状態変数を管理しています．この両方のスロット・カウンタは各コミュニケーション・サイクルの開始時に‘1’に初期化され，各スロットの終了境界で＋1（インクリメント）されます．

静的スロットの番号（gNumberOfStaticSlots）は，

gNumberOfStaticSlots＝2－cStaticSlotIDMax

でセットされます．なお，cStaticSlotIDMaxは静的スロットIDの最高位であり，1,023の値をとります．したがって，静的スロットの最大割り当て数は1,021になります．

図8は，チャネルA，チャネルBの二つを送信する際のすべてのパターンを表しています．例えば静的スロット1の場合，両方のチャネルにフレームが送信されます．また，静的スロット2ではチャネルAにだけフレームを送信し，静的スロット3ではどちらのチャネルにもフレームは送信されていません．

前述したように，静的スロットは各コミュニケーション・サイクルの開始で‘1’に初期化されますが，これはスロット番号とフレームIDが一致するようにするためです．‘0’に初期化されないことにも意味があります．フレームIDは11ビットありますが，「‘0’は無効」と定義されているのです．

静的セグメント内のすべてのコミュニケーション・スロット注3は静的にコンフィグレーションされ，そのスロットの時間は等しく，フレームの長さも等しくなります．さらに静的スロットは以下の制約を持っています．

- 同期フレームは接続されているすべてのチャネルに送信

注3：コミュニケーション・スロットは，実際に通信を行う（フレームを送信する）スロットのこと．静的セグメントでは静的スロットを，動的セグメントではミニスロットを指す．

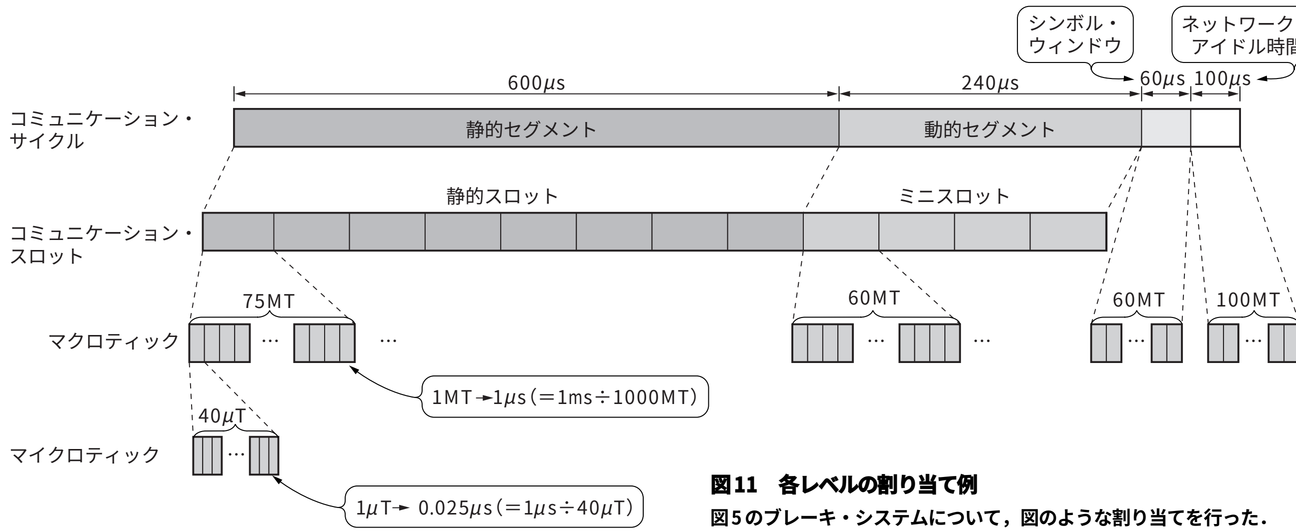


図11　各レベルの割り当て例
図5のブレーキ・システムについて，図のような割り当てを行った．

されるものとする

- 同期フレーム以外のフレームは，A，Bどちらかのチャネル，または両方のチャネルに送信される．
- 一つのノードだけが所定のチャネルに対して所定のフレームID で送信するものとする
- クラスタがシングル・スロット・モード注4に設定されたときは，すべての非同期ノードはシングル・スロット・フレームとしてフレームを明示するものとする．

なお，**図9**に示すFlexRayのフレーム・フォーマットにおいて，同期フレーム・インジケータが 1 ’となったとき，そのフレームを同期フレームとします．

図10に，動的セグメントの構造を示します．静的セグメントとは異なり，フレーム長の変化に応じてコミュニケーション・スロット（ミニスロット）も変わります．ミニスロット数は最大7,994と規定されています（NumberOf Minislots=7994）．

ここで注意しなければならないことがあります．それは，静的セグメントと動的セグメントを通じてのID番号の最高位（cSlotIDMax）が2,047であるということです．つまり，ミニスロット数が最大7,994だとしても，最高位のID番号の2,047で規制されてしまいます．

3）マクロティック・レベル

マクロティック・レベルは，同期サービスを提供する機能を備えています．構成要素はマクロティック（MT）と呼ばれます．一つの静的スロットにおけるマクロティック設定範囲は5MT～2,047MT（gdStaticslot=5...2047MT）と決められています．また，ミニスロットでは2MT～63MT（gdMinislot=2...63MT），シンボル・ウィンドウでは0MT～87MT，ネットワーク・アイドル時間では2MT～773MTと規定されています．

4）マイクロティック・レベル

マクロティック・レベルをさらに細かく分割したのがマイクロティック・レベルです．構成要素はマイクロティック（μT）と呼ばれます．1マクロティックを構成するマイクロティックの公称値は40μT～240μTになっています（pMicroPerMacroNom＝［cMicroPerMacroNomMin－240］μT）．1μTは1，2，4サンプルで構成されます（pSamples PerMicrotick＝［1, 2, 4］）．仕様では，サンプリング・クロック時間が0.0125μs～0.05μs（gdSampleClockPeriod＝0.0125...0.05μs）と規定されていますが，これはFlexRay通信コントローラの動作周波数に関係します．

● 通信コントローラの動作周波数を決める

さて，ここからは**図5**のブレーキ・システムを例にとって，上記のメディア・アクセスの割り当てを行っていきましょう．

図5のブレーキ・システムでは八つのノードを配置しました．また，コミュニケーション・サイクル時間を1msとしました．次に，以下のようにコミュニケーション・サイクル・レベルの各セグメントの時間を割り当てることにしましょう（**図11**）．これを合計すると1,000μs（1ms）になります．

注4：シングル・スロット・モードは，スタートアップのプロセスで使用されるモード（pSingleSlotEnabled＝Boolean）．

- 静的セグメント＝600μs
- 動的セグメント＝240μs
- シンボル・ウィンドウ＝60μs
- ネットワーク・アイドル時間＝100μs

なお，上記に示した数値は説明をわかりやすくするために設定したものであり，実際のブレーキ・システムに基づく値ではありません．実際のBrake-by-wireアプリケーションでは，走行速度や制動距離（ブレーキが利き始めてから自動車が停止するまでの距離），ブレーキ量の伝達などを考慮してネットワークのリアルタイム性を考えなければなりません（このあたりの仕様は自動車メーカによって異なるので，一概には言えない）．いずれにしても，現在の機械制御によるブレーキ・システムと同等かそれ以上の性能を確保できるような設計が求められると思われます．

次にアービトレーション・レベルの割り当てを行いましょう．**図5**のブレーキ・システムには8ノード接続されているので，ここではわかりやすくするために静的スロットを8スロット用意することにします．ここではノード数とスロット数を同数にしましたが，一つのノードが複数のスロットを使うことは珍しいことではありません．次に動的セグメントですが，これはネットワーク管理やノードの診断などに使用されると思われます．今回は，簡単化のため4スロットとします．

次はマクロティック・レベルですが，これを以下のように割り当てます．なお，この数値も説明のために設定したもので，実アプリケーションとは関係はありません．

- 1静的スロット＝75MT
- 1ミニスロット＝60MT
- シンボル・ウィンドウ＝60MT
- ネットワーク・アイドル時間＝100MT

ここまでの割り当てが，1コミュニケーション・サイクル当たりのマクロティック（gMacroPerCycle（12...16,000 MT））の範囲に入っているのかを検証しましょう．

静的セグメント＝8スロット×75MT	＝600MT
動的セグメント＝4スロット×60MT	＝240MT
シンボル・ウィンドウ	＝ 60MT
ネットワーク・アイドル時間	＝100MT
	合計1,000MT

1コミュニケーション・サイクルで1,000MTなので規定範囲内です．また，コミュニケーション・サイクル＝1msと設定したので，1MT＝1ms÷1,000＝1μsとなります．

次に1MTをマイクロティック・レベルで割り当てましょう．1MT当たり40μT（cMicroPerMacroNomMin=40μT）とすると，1μs÷40＝0.025μsとなります．つまり，1μT＝0.025μsです．

前述したように，マイクロティックの構成単位はクロック・サンプルです．この単位は1，2，4です．これを動作周波数に換算すると，

- pSamplesPerMicrotick＝1：40MHz（0.025μs）
- pSamplesPerMicrotick＝2：80MHz（0.0125μs）

pSamplesPerMicrotick＝4の場合はサンプル・クロック時間が0.00625μsでプロトコル違反になり，設定できません．このことから，最小40MHz，最大80MHzのクロック周波数で通信コントローラを動作させることになります．

＊　＊　＊

今回はリアルタイム性を求めるにあたり，メディア・アクセス方式について説明しました．グリッド・レベルのところで説明した静的セグメントで8静的スロットを割り当て，その1スロット時間を75MTとし，1MTを1μsと定義しました．

次回は，このスロット時間に送信されるフレーム・フォーマットについて詳しく説明します．フレーム送信とその受信のメカニズム，ビット・レートの設定についてお話しします．

さとう・みちお
フリースケール セミコンダクター ジャパン（株）

<筆者プロフィール>
佐藤道夫．1982年にモトローラに入社し，半導体事業部に所属．品質保証や営業を経験し，1992年からコントロール＆ネットワークのアプリケーション・エンジニアとして，ロンワークスの普及に従事する．現在は，フリースケール セミコンダクターでトランスポーテーション＆スタンダードプロダクツグループにおいて，おもに車載向けのネットワークの技術担当マネジャをしている．